METHOD, SYSTEM, AND PROGRAM FOR ASSIGNING PRIORITIES

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention is directed to assigning priorities to requests that manipulate (e.g., update or copy) data.

2. Description of the Related Art

[0002] Disaster recovery systems typically address two types of failures, a sudden catastrophic failure at a single point in time or data loss over a period of time. In the second type of gradual disaster, updates to volumes on data storage may be lost. To assist in recovery of data updates, a copy of data may be provided at a remote location. Such dual or shadow copies are typically made as the application system is writing new data to a primary storage device at a primary storage subsystem. The copies are stored in a secondary storage device at a secondary storage subsystem.

[0003] Typically, priorities are assigned to requests that manipulate (e.g., update or copy) data in volumes stored in the primary storage device. The requests may also be referred to as I/O requests. Since the priority assigned to each request is used by a resource manager at the primary and secondary storage subsystems to govern how resources (e.g., processor power for processing the I/O requests, memory for storing data, and hardware to perform data movement) should be allocated to execute a request, if the same request is processed in the primary and secondary storage subsystems with different priorities, the resources in both the primary may not be efficiently managed and can cause resource constraints.

[0004] For example, if the primary storage subsystem handles certain requests with high priority, and the secondary storage subsystem handles the same requests with a lower priority, these requests may not be able to finish within a reasonable amount of time because they will be waiting for the secondary storage subsystem to complete requests with higher priority.

Thus, even though the most of the resources in the primary storage subsystem are dedicated to handle these requests, they are treated differently at the secondary storage subsystem.

[0005] On the other hand, when the primary storage subsystem handles certain requests with low priority, and the secondary storage subsystem handles these requests with a higher priority, the secondary storage subsystem may not have enough resources to finish higher priority requests from the primary storage subsystem in a reasonable amount of time.

[0006] Thus, there is a need in the art for storage subsystems to effectively manage their resources to reduce I/O response time and increase overall system throughput.

SUMMARY OF THE INVENTION

[0007] Provided are a method, system, and program for assigning priorities. A request to manipulate data is received. A type of the request is determined. A priority is assigned to the request based on the type of the request.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 illustrates, in block diagrams, a computing environment in accordance with certain implementations of the invention.

FIGs. 2A, 2B, and 2C illustrate logic implemented in a priority assignment process at a primary control unit in accordance with certain implementations of the invention.

FIGs. 3A, 3B, and 3C illustrates ranges of priorities in accordance with certain implementations of the invention.

FIG. 4 illustrates logic implemented at a secondary control unit in accordance with certain implementations of the invention.

FIG. 5 illustrates one implementation of the architecture of computer systems in accordance with certain implementations of the invention.

DETAILED DESCRIPTION

[0008] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several implementations of the present invention. It is understood that other implementations may be utilized and structural and operational changes may be made without departing from the scope of the present invention.

[0009] Implementations of the invention apply consistent priorities to I/O requests at both a primary and a secondary control unit (e.g., storage subsystems).

[0010] FIG. 1 illustrates, in block diagrams, a computing environment in accordance with certain implementations of the invention. A primary control unit 100 provides one or more hosts (e.g., host 114) access to primary storage 112, such as Direct Access Storage Device (DASD). The primary storage 112 may be divided into blocks of storage containing blocks of data, and the blocks of storage are further divided into sub-blocks of storage that contain sub-blocks of data. In certain implementations, the blocks of data are contents of tracks, while the sub-blocks of data are contents of sectors of tracks. For ease of reference, the terms tracks and sectors will be used herein as examples of blocks of data and sub-blocks of data, but use of these terms is not meant to limit the technique of the invention to tracks and sectors. The techniques of the invention are applicable to any type of storage, block of storage or block of data divided in any manner.

[0011] The primary control unit 100 includes a primary cache 116 in which updates to blocks of data in the primary storage 112 are maintained until written to primary storage 112 (e.g., tracks are destaged). Primary cache 116 may be any type of storage, and the designation of cache illustrates only certain implementations. Additionally, the primary control unit 100 includes a nonvolatile cache 118. The non-volatile cache 118 may be, for example, a battery-backed up volatile memory, to maintain a non-volatile copy of data updates.

[0012] The primary control unit 100 may include one or more copy processes 102 (e.g., for executing an establish with copy command), one or more async processes (e.g., for executing an Peer-to-Peer Remote Copy (PPRC) Extended Distance or asynchronous PPRC copy command), and one or more sync processes 106 (e.g., for executing a synchronous PPRC copy command). Each of the processes 102, 104, and 106 transfers data from the primary control unit 100 to remote storage, such as storage at the secondary control unit 120. In certain implementations, the async process 104 runs continuously for PPRC Extended Distance and asynchronous PPRC commands, and the synch process 106 starts up and completes for a synchronous PPRC command. In certain implementations, there may be different async processes 104 for asynchronous PPRC and for PPRC Extended Distance).

[0013] International Business Machines Corporation (IBM), the assignee of the subject patent application, provides several remote mirroring systems, including, for example: a synchronous PPRC service, an asynchronous PPRC service, a PPRC Extended Distance service, or an establish with copy command in an Enterprise Storage Server® (ESS) system. For ease of reference, the synchronous Peer-to-Peer Remote Copy (PPRC) service, asynchronous PPRC service, and PPRC Extended Distance service will be described as providing synchronous PPRC, asynchronous PPRC, and PPRC Extended Distance commands.

[0014] The synchronous PPRC service provides a technique for recovering data updates that occur between a last, safe backup and a system failure with a synchronous PPRC command. Such data shadowing systems can also provide an additional remote copy for non-recovery purposes, such as local access at a remote site. With the synchronous

PPRC service, a primary storage subsystem maintains a copy of predefined datasets on a secondary storage subsystem. The copy may be used for disaster recovery. Changes to data are copied to the secondary storage subsystem as an application updates the data. Thus, the copy may be used whether there are gradual and/or intermittent failures. The copy is maintained by intercepting write instructions to the synchronous PPRC dataset and generating appropriate write instructions from the primary storage system to the secondary storage system. The write instructions may update data, write new data, or write the same data again.

[0015] The synchronous PPRC service copies data to the secondary storage subsystem to keep the data synchronous with a primary storage subsystem. That is, an application system writes data to a volume and then transfers the updated data over, for example, Enterprise System Connection (ESCON®) fiber channels to the secondary storage subsystem. The secondary storage subsystem writes the data to a corresponding volume. Only when the data is safely written to volumes at both the primary and secondary storage subsystems does the application system receive assurance that the volume update is complete.

[0016] Thus, with synchronous PPRC commands, the copy at the secondary storage subsystem is maintained by intercepting write instructions to the dataset at the primary storage subsystem and generating appropriate write instructions from the primary storage system to the secondary storage system.

[0017] For synchronous PPRC, before the host 114 receives an acknowledgment of completion of the write process when writing a chain of tracks to the primary control unit 100, all tracks in the chain are also transferred to the secondary control unit 120 by a sync process 106.

[0018] Asynchronous PPRC and PPRC Extended Distance commands do not write to secondary storage subsystem before acknowledging the write to the primary storage subsystem. Instead, for the PPRC Extended Distance service, when a block of data is written, information is stored that indicates that the block of data is to be transferred to the secondary storage subsystem at a later time. An asynchronous process collects updates at the primary storage subsystem and sends the updates to the secondary storage subsystem.

[0019] For PPRC Extended Distance, the host 114 may complete writing a track to the primary control unit 100 without the track having been sent to the secondary control unit 120. After the track has been written to the primary control unit 100, the sync process 106 will discover that an indicator corresponding to the track is set to indicate that the track is out of sync with a corresponding track at the secondary control unit 120 and will send the track to the secondary control unit 120. That is, the track is sent asynchronously with respect to the track written by the host.

[0020] With an establish with copy command, a copy of a volume at the primary storage subsystem is made at the secondary storage subsystem during an initial copy relationship. After this, updates made to the volume at the primary storage subsystem may be copied to the corresponding copy of the volume at the secondary storage subsystem to keep the copies of the volume in sync.

[0021] The primary control unit 100 also includes one or more resource management processes 108 for managing resources and a priority assignment process 110 for assigning priorities to I/O requests.

[0022] In certain implementations, the processes 102, 104, 106, 108, and 110 are implemented as firmware. In certain implementations, the processes 102, 104, 106, 108, and 110 are implemented in a combination of firmware and software. In certain implementations, the processes 102, 104, 106, 108, and 110 are implemented as separate software programs for each process 102, 104, 106, 108, and 110. In certain implementations, the processes 102, 104, 106, 108, and 110 may be combined with each other or other software programs (e.g., the async processes 104 and sync processes 106 may be combined with each other).

[0023] Channel adaptors 140A. . . 140N allow the primary control unit 100 to interface to channels. For ease of reference, A...N are used to represent multiple components (e.g., 140A. . . 140N). In certain implementations, channel adaptors 140A. . . 140N may be Fibre channel adaptors.

[0024] Secondary control unit 120 allows access to disk storage, such as secondary storage 122, which maintains back-up copies of all or a subset of the volumes of the primary storage 112. Secondary storage may be a Direct Access Storage Device (DASD). Secondary storage 122 is also divided into blocks of storage containing blocks of data, and the blocks of storage are further divided into sub-blocks of storage that contain sub-blocks of data. In certain implementations, the blocks of data are tracks, while the sub-blocks of data are sectors of tracks. For ease of reference, the terms tracks and sectors will be used herein as examples of blocks of data and sub-blocks of data, but use of these terms is not meant to limit the technique of the invention to tracks and sectors. The techniques of the invention are applicable to any type of storage, block of storage or block of data divided in any manner.

[0025] The secondary control unit 120 also includes one or more resource management processes 128 for managing resources and a priority assignment process 130 for assigning priorities to I/O requests. In certain implementations, the processes 128 and 130 are implemented as firmware. In certain implementations, the processes 128 and 130 are implemented in a combination of firmware and software. In certain implementations, the processes 128 and 130 are implemented as separate software programs for each process 128 and 130. In certain implementations, the processes 128 and 130 may be combined with each other or other software programs.

[0026] Channel adaptors 150A. . . 150N allow the secondary control unit 120 to interface to channels. For ease of reference, A...N are used to represent multiple components (e.g., 150A. . . 150N). In certain implementations, channel adaptors 150A. . . 150N may be Fibre channel adaptors.

[0027] Although for ease of illustration, only a communication paths 170 and 172 are illustrated, there may be communication paths between host 114 and each channel adapter 140A... 140N and between channel adapters 140A... 140N and channel adapters 150A... 150N.

[0028] In certain implementations, communication path 172 between channel adapter 140N and 150A is bidirectional. Also, either control unit 100 or 120 may be designated a primary control unit, and the other control unit may be designated as a secondary control unit for certain commands. For example, control unit 100 may be designated as a primary control unit 120 for an asynchronous PPRC command, while control unit 120 may be designated as a primary control unit 120 for an establish with copy command (e.g., to make an initial copy of a volume).

[0029] Thus, a channel adaptor 140A... 140N may receive I/O requests from communication path 170 or communication path 172. In certain implementations, the I/O requests may include, for example, host I/O commands, asynchronous PPRC commands, Extended Distance PPRC commands, synchronous PPRC commands, and establish with copy commands. Implementations of the invention assign priorities to each of these I/O requests with the priority assignment processes 110 and 130.

[0030] The priority assignment processes 110 and 130 assign a priority to each I/O request based on the I/O type of the I/O request. In certain implementations, the I/O type refers to whether the I/O request is a host I/O command, an asynchronous PPRC command, an Extended Distance PPRC command, a synchronous PPRC command, or an establish with copy command. However, any type of I/O request falls within the scope of the invention.

[0031] In certain implementations, the primary control unit 100 and secondary control unit 120 communicate via communication paths, such as direct high speed transmission lines (e.g., an Enterprise System Connection (ESCON®) link). However, the communication paths may be comprised of any other communication means known in the art, including network transmission lines, fiber optic cables, etc., as long as the primary control unit 100 and secondary control unit 120 are able to communicate with each other.

[0032] In certain implementations, the primary control unit 100 and secondary control unit 120 may be comprised of the IBM® 3990, Model 6 Storage Controller, Enterprise Storage Server®, or any other control unit known in the art, as long as the primary control unit 100 and secondary control unit 120 are able to communicate with each other.

[0033] In certain implementations, the primary control unit 100 and/or secondary control unit 120 may comprise any computing device known in the art, such as a mainframe, server, personal computer, workstation, laptop, handheld computer, telephony device, network appliance, virtualization device, storage controller, etc.

[0034] A primary site may include multiple primary control units, primary storage, and host computers. A secondary site may include multiple secondary control units, and secondary storage.

[0035] In certain implementations of the invention, data is maintained in volume pairs. A volume pair is comprised of a volume in a primary storage device (e.g., primary storage 112) and a corresponding volume in a secondary storage device (e.g., secondary storage 122) that includes a consistent copy of the data maintained in the primary volume. For example, primary storage 112 may include VolumeA and VolumeB, and secondary storage 122 may contain corresponding VolumeX and VolumeY, respectively.

[0036] In certain implementations, removable and/or non-removable storage (instead of or in addition to remote storage, such as secondary storage 122) may be used to maintain back-up copies of all or a subset of the primary storage 112, and the techniques of the invention transfer data to the removable and/or non-removable storage rather than to the remote storage. The removable and/or nonremovable storage may reside at the primary control unit 100.

[0037] FIGs. 2A, 2B, and 2C illustrate logic implemented in a priority assignment process 110 at a primary control unit 100 in accordance with certain implementations of the invention. For each volume at the primary storage subsystem, there is a corresponding volume at the secondary storage subsystem. At any given time, synchronous copy commands, asynchronous copy commands, and establish with copy commands may be executing concurrently. The priority assignment process 110 assigns a priority based on the I/O type of the I/O request. The priority is used to ensure that host I/O response time is not impacted by establish with copy I/O requests. The priority is also used to prevent synchronous and asynchronous PPRC I/O requests with higher priority from starving establish with copy I/O requests with lower priority.

[0038] In FIG. 2A, control begins at block 200 with the priority assignment process 110 receiving an I/O request. In block 202 with the priority assignment process 110 determines whether the I/O request was issued with a synchronous PPRC command. If so, processing continues to block 204, otherwise, processing continues to block 210.

[0039] In block 204, the priority assignment process 110 determines whether a host 114 assigned a priority to the I/O request. If so, processing continues to block 206, otherwise, processing continues to block 208. That is, a host 114 may assign each I/O request a priority to be applied to the target of the host for synchronous PPRC.

[0040] The host 114 assigns a priority from a range of priorities. FIG. 3A illustrates ranges of priorities used by host 114 in accordance with certain implementations of the invention. Range 300 represents possible priority values that a host 114 may assign to an I/O request. In this example, the values range from 1 to 4, with 1 being the highest priority. FIG. 3B illustrates ranges of priorities used by primary control unit 100 in accordance with certain implementations of the invention. Range 310 represents possible priority values that the priority assignment process 110 may assign to an I/O request. In this example, priorities 1, 2 and 3 are in the high priority range; priorities 4 and 5 are in the medium priority range; and, priorities 6 and 7 are in the low priority range.

[0041] In block 206, the priority assignment process 110 maps the host priority to a priority within a high priority range. With reference to the example of FIGs. 3A and 3B, the priority assignment process 110 would map any I/O request from host 114 in the high priority range, but the priority assignment process 110 takes into account the host priority and, for example, pending I/O requests and resources at the primary control unit 100, when mapping to a particular priority value (e.g., 1, 2, or 3). For example, if an establish with copy command was being performed for data that the host I/O request was attempting to update, that copy command would have to be completed before the host I/O request is processed. Therefore, in this case, the host I/O request may be assigned a priority of 3. Additionally, if needed, the priority of the establish with copy command may be increased so that the establish with copy command finishes more quickly (block 218).

[0042] In block 208, the priority assignment process 110 assigns a priority from the high priority range to the I/O request. In block 210, the priority assignment process 110 determines whether the I/O request is for an I/O request issued with an establish with copy command. If so, processing continues to block 212, otherwise, processing continues to block 214. In block 212, the priority assignment process 110 assigns the I/O request a priority from the low priority range based, for example, on pending I/O requests and available resources at the primary control unit 100.

[0043] In block 214, for a request issued with an asynchronous PPRC command or and Extended Distance PPRC command, the priority assignment process 110 assigns the I/O request a priority from the medium priority range based, for example, on pending I/O requests and available resources at the primary control unit 100.

[0044] In block 216, the priority assignment process 110 sends the I/O request the priority it has assigned and a host priority (if one exists for the I/O request) to the secondary control unit 120. In certain implementations, the priority assignment process 110 stores the priority it has assigned and a host priority (if one exists for the I/O request) in an extended portion of a Command Descriptor Block (CDB), which is embedded in a Fibre Channel Protocol (FCP) command to be sent to the secondary control unit 120.

[0045] In block 218, the priority assignment process 110 optionally updates priorities of pending I/O requests as needed. In block 220, the priority assignment process 110 performs other processing.

[0046] Both the primary control unit 100 and the secondary control unit 120 allocate resources to handle a request based on the host priority and a priority assigned by the primary control unit 100.

[0047] FIG. 4 illustrates logic implemented at a secondary control unit 120 in accordance with certain implementations of the invention. Control begins at block 400 with the secondary control unit 120 receiving an I/O request with a priority assigned by the priority assignment process 110 at the primary control unit 100 and including a host priority if one exists for the I/O request. In block 402, the secondary control unit 120 processes the I/O request using the priority assigned by the priority assignment process 110 at the primary control unit.

[0048] Thus, implementations of the invention provide a technique to apply consistent priorities for handling I/O requests in a primary control unit 110 and secondary control unit 120 by assigning priorities based, at least in part, on I/O types of the I/O requests. Also, for synchronous PPRC, since consistent priority also applies to the host 114 and target (e.g., one or more blocks of data in primary storage 112), the primary and secondary control units 110 and 120 incorporate the priority assigned by the host 114 in handling an I/O request. With consistent priorities applied to all involved control units 110 and 120, I/O response time is reduced and overall system throughput is increased.

[0049] IBM, Enterprise Storage Server, and ESCON are registered trademarks or common law marks of International Business Machines Corporation in the United States and/or foreign countries.

Additional Implementation Details

[0050] The described techniques for assigning priorities may be implemented as a method, apparatus or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks,, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. The code in which various implementations are implemented may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Thus, the "article of manufacture" may comprise the medium in which the code is embodied. Additionally, the "article of manufacture" may comprise a combination of hardware and software components in which the code is embodied, processed, and executed. Of course, those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention, and that the article of manufacture may comprise any information bearing medium known in the art.

[0051] The logic of FIGs. 2A, 2B, 2C, and 4 describes specific operations occurring in a particular order. In alternative implementations, certain of the logic operations may be performed in a different order, modified or removed. Moreover, operations may be added to the above described logic and still conform to the described implementations. Further, operations described herein may occur sequentially or certain operations may be processed in parallel, or operations described as performed by a single process may be performed by distributed processes.

[0052] The illustrated logic of FIGs. 2A, 2B, 2C, and 4 may be implemented in software, hardware, programmable and non-programmable gate array logic or in some combination of hardware, software, or gate array logic.

[0053] FIG. 5 illustrates an architecture 500 of a computer system that may be used in accordance with certain implementations of the invention. Host 114, primary control unit 100, and/or secondary control unit 120 may implement computer architecture 500. The computer architecture 500 may implement a processor 502 (e.g., a microprocessor), a memory 504 (e.g., a volatile memory device), and storage 510 (e.g., a non-volatile storage area, such as magnetic disk drives, optical disk drives, a tape drive, etc.). An operating system 505 may execute in memory 504. The storage 510 may comprise an internal storage device or an attached or network accessible storage. Computer programs 506 in storage 510 may be loaded into the memory 504 and executed by the processor 502 in a manner known in the art. The architecture further includes a network card 508 to enable communication with a network. An input device 512 is used to provide user input to the processor 502, and may include a keyboard, mouse, pen-stylus, microphone, touch sensitive display screen, or any other activation or input mechanism known in the art. An output device 514 is capable of rendering information from the processor 502, or other component, such as a display monitor, printer, storage, etc. The computer architecture 500 of the computer systems may include fewer components than illustrated, additional components not illustrated herein, or some combination of the components illustrated and additional components.

[0054] The computer architecture 500 may comprise any computing device known in the art, such as a mainframe, server, personal computer, workstation, laptop, handheld computer, telephony device, network appliance, virtualization device, storage controller, etc. Any processor 502 and operating system 505 known in the art may be used.

[0055] The foregoing description of implementations of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many implementations of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.